On Generating High Quality "Water-tight" Triangulations Directly from CAD

Robert Haimes
Massachusetts Institute of Technology
haimes@mit.edu

and

Michael J. Aftosmis
NASA Ames Research Center
aftosmis@nas.nasa.gov

Abstract

This paper presents a robust technique for generating water-tight triangular surface tessellations directly coupled to CAD geometry. These triangulations can match user specified criteria for chord-height tolerance, neighbor triangle dihedral angle, and maximum edge length. This discrete representation has *hooks* back to the owning geometry and therefore can be used in conjunction with these entities to allow for easy enhancement or modification of the tessellation suitable for grid generation or other downstream applications.

1 Introduction

A paramount design goal for the **C**omputational **A**nalysis **PR**ogramming **I**nterface (**CAPRI**) [1] is that geometry access be appropriate for developers of grid generators, solvers and visualization subsystems. The complete Computational Geometry (CG) perspective was avoided while maintaining full functionality. The simplification of the data definition and Application Programming Interface (API) provides ease of software generation without regard for special cases. For example, **CAPRI** does not expose the type of a geometric entity but only that it is a parameterized object with certain bounds. This allows for uniform access regardless of form. Periodic curves are broken so that they are always bound by two endpoints.

Other important **CAPRI** design goals include:

- Support Manifold Solids. By only supporting solid geometry, the problems in trimming surfaces do not exist. If handled properly, the geometry need not be fixed or modified. Data readers can be truly hands-off.
- Direct CAD Access. The data exposed through **CAPRI** exists in the CAD system. There is no geometry translation, thus avoiding the errors

and other problems associated with CAD model translation.

- Provide Boolean Solid Operators. This allows for multi-disciplinary analysis. If the geometry exists for a blade, a subtraction from a wedge can be performed to produce a passage. Therefore, the same definition can easily be used for both the fluids and structures.
- Master-Model manipulation. By allowing the specification of both the parameters values (that define the geometry construction) and suppression of nodes of the "feature-tree" different instances of the part (or assembly) can be constructed. This portion of **CAPRI** allows for both geometric parameter studies as well as full design optimization.

1.1 CAPRI's Architecture

Instead of the traditional serial methods to analysis, **CAPRI** uses a geometry centric approach. The connection to the geometry is made through an API - **NOT** a file system. The API isolates the top-level applications (grid generators, solvers and visualization components) from details of the geometry engine. This architecture allows for the simple replacement of one geometry kernel/CAD system with another without effecting the top-level **CAPRI** applications. Figure 1 shows this relationship schematically where **CAPRI** is positioned as middleware between user applications and the geometry kernel and database.

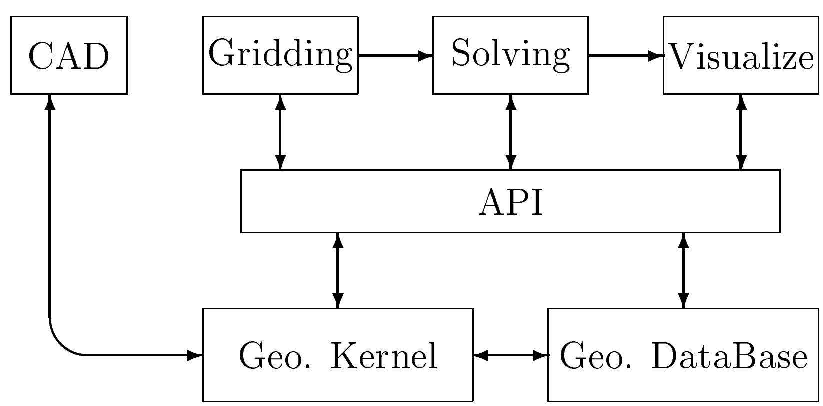

Figure 1: The **CAPRI**-based Computational Analysis Suite

CAPRI requires two primary definitions; (1) the API itself with bindings to C++, C and FORTRAN and (2) an object data definition which, used through the API, communicates the information in a bi-directional manner.

1.2 Computational Geometry Definitions

When dealing with solid entities CG distinguishes between topology and geometry of the bodies. *Geometry* refers to the shape of the object in physical or paramter space, and CAD solids usually do this through an implicit mathematical relations [3]. *Topology* refers to the connectivity, hierarchy, bounds, and relative orientation between entities (either topological or geometric). Geometric surface queries yield coordinates in (x, y, z) or (u, v), and topologic queries produce results that are almost always true/false. Some topological entities have underlying geometry; others are collections of other topologic entities. A table of the full CG hierarchy for CAD solids can be seen in Figure 2.

Topologic entity	Geometric entity	Parameterization
Body		
Region		
Shell		
Face	surface	$(x, y, z) = f(u, v)$
Loops		
Edge	curve	$(x, y, z) = g(t)$
Node	point	

Figure 2: Computational Geometry entities

CAPRI does not expose Regions and Shells through the API. These objects can be determined by traversing through the Faces and examining the neighbors (manifolds have the property that any Edge points to exactly two Faces). **CAPRI** also does not make the distinction between topology and geometry and exposes only Body as the main *container* which holds Faces, Edges and Nodes. Loops exist only as part of the owning Face data. The Face API routines expose the (u, v) parameterization of the surface and the Edge routines expose only the running parameter, (t), of the curve.

1.3 CAD Representation of Geometry

CAD systems have a tolerance that represents "closure" for solids. This means that the Nodes that bound an Edge are probably not on the underlying curve, and the Edges that bound a Face (through the Loops) are not necessarily on the supporting surface. All that is required is that the bounding objects be within a specified tolerance. Therefore at any precision higher than the tolerance, gaps and overlaps may exist in the topology/geometry definition. This tolerance is generally much larger than those associated with double precision floating-point arithmetic (*e.g.* the default relative tolerance for Pro/ENGINEER is only 10^{-2}).

In order to deal with the gap and overlaps, most CAD-based applications must "fix" the geometry. This usually entails translating the geometric definition to another (simpler) representation where the bounding entities fall closer to, or on the object. This type of translation has a variety of side effects, including:

- Inconsistency: Not querying the same geometry. Since the geometry has changed, the representation is different than in the CAD system.
- Complexity: At times additional Faces are required to close the model. There is no way to predict how many of these "sliver faces" may need to

be introduced, moreover, slivers can cause problems for grid generators.

CAPRI's perspective is that the geometry in the CAD system is *truth* and should not be modified (though **CAPRI** may modify the topology). Therefore fixing the CAD's model is not an option. This is where the work here differs fundamentally from Gerteisen and Baker[2].

1.4 Evolution of CAPRI Triangulations

Early in the design and implementation of **CAPRI**, it became obvious that providing an API that gives the programmer access to the geometry and topology of a solid part was not sufficient. The burden of deciphering the CAD data and attempting to generate a discrete representation of the surface required for mesh generation was too great for most gridding subsystems. This became apparent when comments about early releases of **CAPRI** were analyzed. Many grid generation systems (used in CFD and other disciplines) can use STL (*Stereo Lithography*) files as a start. Combining a discretized view of the solid part as well as it's geometry and topology can provide a complete, and easier to use, access point into the CAD data. A tessellation of the object that contains not only the coordinates and supporting triangle indices but other data such as the surface parameters (for the points) as well as the connectivity of the triangles assists the investigator in traversing through and dissecting the CAD representation of a part.

An important aspect of **CAPRI** is that it provides CAD vendor neutral access to all of the data it passes back to the application. This burden is high when providing a complete, correct and "water-tight" tessellation. Some CAD system geometry kernels can provide data of this quality (i.e., UniGraphics through Parasolid, CATIA and ComputerVision). Other CAD interfaces can provide the data but it is not of sufficient enough quality to use (for example, Pro/Engineer). Finally, SDRC's Open I-DEAS API does not provide access to a triangulation at all.

The fact that not all CAD systems provide such a tessellation has forced the development of a surface triangulator within **CAPRI** for CAD solid parts that does meet all of **CAPRI**'s requirements.

Subsequent **CAPRI** releases provided a correct (geometrically and logically) triangulation of the part for PTC's Pro/ENGINEER and SDRC's I-DEAS. The triangulation technique began with a discretization of the bounding Edges and attempted to tessellate the solid one Face at a time. Starting from the Edges ensured that the result was water-tight. This triangulator relied upon various *ad hoc* schemes including point-peppering algorithms to form triangles with the uniform orientation. Enough different schemes (including hierarchical quad-tree and curvature based methods) were used so that the failure rate was very low. As with many *ad hoc* methods, however, this situation was never fully satisfying since it could not always ensure success and infrequent situations arose in which surfaces could not be represented.

CAPRI's tessellations were not intended as an appropriate starting point for any computational analysis. **CAPRI** sees only geometry, and it cannot anticipate the smoothness, resolution or other requirements of the downstream application(s). The triangulations describe the geometry only, and some processing of the tessellation is expected in order to get the triangulation suitable for the physical problem being investigated. At that time, the triangulation can be enhanced through either physical or parameter space manipulation, using (u, v) evaluations and point "snap" routines provided by **CAPRI** [5].

In order to improve upon the quality and robustness of the triangulations provided by early releases of **CAPRI** the *ad hoc* techniques had to be scrapped. A new method needed to be developed with the following goals:

- Robust. This is the most important of the design goals. It is imperative that the scheme always work and provide a tessellation that can be used.
- Correct. The triangulation is of no use if it is not correct. The tessellation must be logically correct – provide a valid triangulation in the parameter space (u, v) of the individual surface. And, it must be geometrically correct – depict a surface triangulation that truly approximates the geometry. This involves having all facets with a consistent orientation and no creases or abrupt changes in triangle normals. Correctness in both physical and parameter space allows downstream enhancement schemes to operate in either or both.
- Adjustable. To minimize the post-processing of **CAPRI**'s tessellation for a specific discipline or analysis, some adjustment to the resultant quality will be supplied in the final method. It must be noted, that any criteria may not be met (especially near the bounds of a CAD object) do to issues of closure and solid model accuracy. This goal may conflict with the more important quality of water-tight and smooth surface representation.
- Minimal number of CAD API queries. Performance issues stress minimization of the total number of direct CAD queries. It can be assumed that any CAD access will be expensive (some systems are client/server and require network communication). This goal conflicts with some of the adjustable criteria. For example, it can be expensive to insure that all triangle midpoints (or sides) do not deviate too far from the actual surface without querying the surface for each of these tests.
- No geometric translation. To truly facilitate hand-off grid generation, anything that requires user intervention must be avoided. All data maintained within **CAPRI** is consistent with the CAD's solid representation. An alternate or translated representation must not be used because then the result will be something different than resides within the CAD. The assumption that a surface or curve parameterization can be accessed and will accurately return coordinates at high precision will be used.
- Water-Tight. This requirement allows meshing without "fixing" geometry. Triangulated CAD solids are closed and conformal, therefore they can be used without intervention. For the tessellation of a solid object that means that all Edge curves terminate at consistent coordinates of the bounding Nodes and a single discretization for Edge curves be used on both Faces that share the Edge. Each side in the triangulation points to exactly two triangles, and the star of each vertex is surrounded by a single closed loop

of sides. The triangulation is everywhere locally *manifold*. In a manifold triangulation, there are no voids, cracks or overlaps of any triangles that make up the solid.

- Periodic Surfaces. As stated before, **CAPRI** internally breaks up periodic surfaces so all parameter values for a point on the curved surface are single valued. Therefore a cylindrical surface would be broken into 2 Faces. Though of lesser importance, the triangulation scheme should not have this same restriction. In this case a cylinder would require only a single seam (at the periodic position) where that Edge would be in the Face's Loop twice, each with a differing set of (u, v)s.

2 Face Tessellations

Most traditional curved surface triangulation schemes require a detailed knowledge of the surface geometry (not just the parameterization) to properly function [4, 5]. The surface knowledge is used to effectively select surface points in an *a priori* manner so that the resultant tessellation scheme closes and is correct.

In this case, with only the parameterization, a different scheme must be developed. It must be able to place points on the surface without the detailed surface knowledge, developing point locations as the triangulation is refined. The outline for the algorithm is:

- Start from Edge discretization. Again, starting from the Edge definitions insures that the resultant complete Body tessellation is water-tight. Of course, the resultant surface tessellation quality is partly a function of the quality of the Edge discretization. An outline for that algorithm can be seen in the next Section.
- Boundary triangulation in (u, v). Theorems in CG state that any planar straight-line graph (PSLG) can be triangulated [6]. Since the Loops (ordered collection of Edges) that bound any Face form a PSLG in the 2-D parameter space, this triangulation is guaranteed. The resulting triangulation can be used as an effective and robust starting point for the Face tessellation.
- Incrementally improve the tessellation. The starting point is correct in (u, v) but probably not very good in (x, y, z), triangles are broken-up in parameter space based on criteria in physical space. This keeps the (u, v) triangulation correct while better approximating the surface. This is not unlike a procedure where one iteratively switches between parameter and physical space [7]. Incremental improvement continues until the centroid of every triangle in (u, v) falls in the same triangle in (x, y, z).
- Improve the tessellation to meet user-defined criteria. The user may specify:
  - Maximum triangle side length. Any triangle sides (not on a CAD Edge) longer (in (x, y, z)) than a specified value are bisected.
  - Maximum dihedral angle between two triangles. Any two triangles on the same CAD Face whose (x, y, z) facet normals differ by more than the input value will be broken up.

– Chord-height tolerance. When the deviation between triangle center and actual surface (in (x, y, z)) is greater than the specified value then the triangle is subdivided by inserting the center point.

An analogy to this process would be to start with a deflated balloon stretched out and supported by the bounding Edges. Then iteratively pin a point at a time to the actual shape while not overlapping the membrane. The ploy is to properly select those points.

2.1 Operations & Methods

There are two types of CAD queries that are useful for point selection. The first is just a parameter evaluation, i.e. $(x, y, z) = f(u, v)$. This has the advantage that it is always precise (to machine accuracy). The second is a "snap", $(u, v) = f^{-1}(x, y, z)$, which tends to be more computationally expensive in that a Newton-Raphson iterative scheme is usually employed by CAD systems to complete the operation. Snaps have other side-effects; (1) because of the numerical scheme there is some granularity to the result (and this should be a function of the models accuracy and is CAD system dependant), (2) care must be taken near periodics so that the result is in the correct quadrant and (3) the result may not be unique. Clearly this is an exaggerated uniqueness example, but consider a point at the center of a sphere – what is the closest point on the surface?

These issues provide motivation for avoiding "snaps" during the Face tessellation phases of the surface mesh generation. Unfortunately, there are no alternatives during the Edge discretization, however, since CAD edges are lower dimensional, only a limited number of "snaps" need to be performed. Edge operations are discussed after the presentation of the Face tessellation methods.

The following triangle based operations are simple procedures that maintain the ability of the collection of triangles to cover the Face area:

2.1.1 (u, v) triangle side bifurcation

This operation cuts up a side of a triangle as can be seen in Figure 3. The result of this new point is 3 additional triangles.

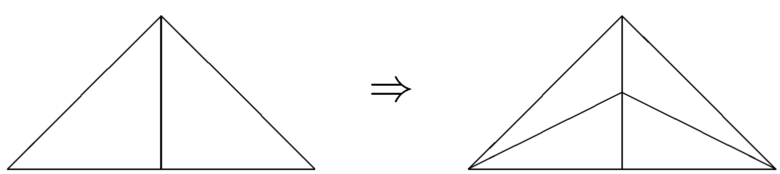

Figure 3: Side Bifurcation Operation

No bifurcation is performed when:

- The side is on an Edge discretization.
- Not closer to an Edge segment than the maximum distance between all Edge points and their projections on the surface.

- The side length is already less than the minimum segment length in the Edge discretization.

2.1.2 (u, v) triangle subdivision

This operation splits up a triangle at its centroid as can be seen in Figure 4. The result of this procedure is the addition of one additional vertex and 2 new triangles.

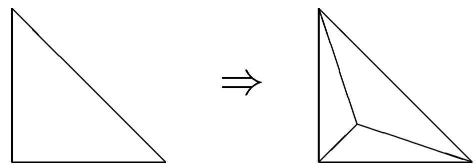

Figure 4: Triangle Subdivision Operation

Triangles are not subdivided when:

- 2 sides of the triangle are Edge segments.
- The midpoint is closer to an Edge segment than the maximum distance between all Edge points and their projections on the surface.
- The center is closer than a fraction of the distance between the points of the nearest Edge segment.

2.1.3 surface recovery (edge-swapping)

This series of local operations is performed on all pairs of triangles that make up the tessellation. Each set is compared based on specific criteria and the decision is made on whether or not to swap the position of the diagonal. The local part of the procedure is depicted in Figure 5 and it should be noted that a pair is only considered for swapping when there is visibility in (u, v) for the diagonal within the triangles. The entire tessellation is iterated upon until no diagonals are swapped.

Three predicates are used for surface recovery.

1. Swap to minimize the maximum angle in (u, v). This drives the triangulation toward MinMax in the (u, v) plane.
2. Swap to minimize the angle between 3D triangle normals. This aligns the diagonals with the principle direction of local surface curvature.
3. Swap to minimize the maximum angle in (x, y, z). This drives the triangulation toward MinMax in physical space.

After any point has been inserted (during any phase of the procedures listed below), two competing reconstruction sweeps are performed. The first sweep minimizes the (u, v) angles then the second minimizes the 3D angles between facet normals. This ensures that the end result both follows the surface curvature and produces MinMax-like triangles.

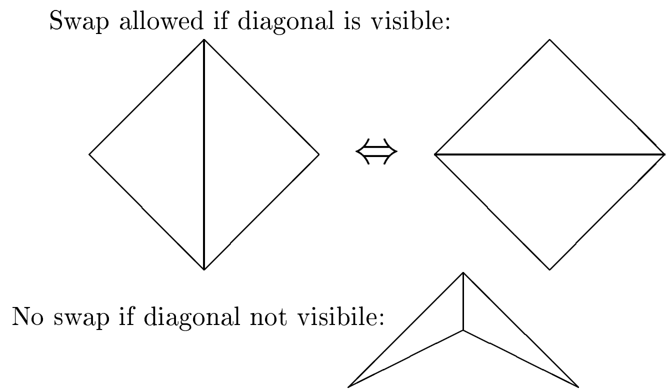

Figure 5: Surface reconstruction via diagonal swapping

2.2 Generation of a Consistent Triangulation

The procedure for Face tessellation is divided into two sections. The first part produces a triangulation that is "consistent" in both (u, v) and (x, y, z) and coarsely approximates the surface. This triangulation is the starting point for the final improvement phases which add additional sites and perform edge-swaps to meet the meshing requirements specified by the user.

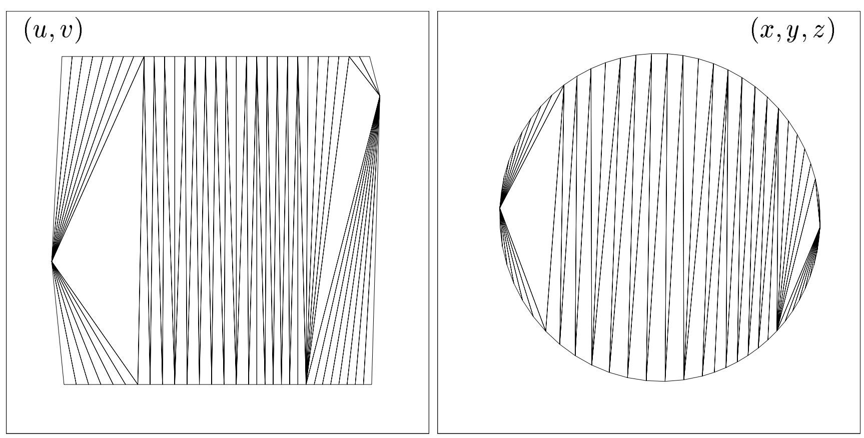

Figure 6: Phase 1 – Initial (u, v) mapping & (x, y, z) for 1/2 sphere. # points = 64, # triangles = 62, # CAD queries = 0.

The consistency phases are as follows:

1. (u, v) Boundary triangulation of CAD Faces. The initial step for any CAD Face is to collect the bounding Edges. The Edge discretizations are ordered (as prescribed by the Loop definitions) to produce one or more closed contours. The triangulation algorithm used is a Collapsing Front technique that is similar to Advancing Front but the only available points are those

that already exist within the front (new sites are not added to the interior of the CAD Face). The smallest segment from the active front is always selected as the site for the next triangle. This continues iteratively until the front is exhausted.

Figure 6 depicts both the parameter and physical space for a hemisphere after the triangulation. Note the poles on both the right and left of the (u, v) plot. These points are singular in the parameter space.

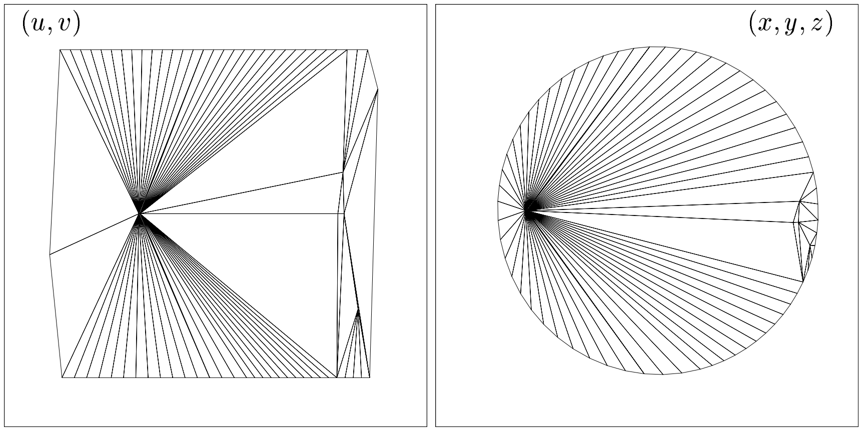

Figure 7: Phase 2 – Split sides where vertices are on different CAD Edges. # points = 70, # triangles = 74, # CAD queries = 9.

2. Split sides where the vertices are on different Edges. The minimum segment distance constraint is not used. This phase ensures that future point insertions are free from this constraint. Figure 7 displays the result. Insertions continue on the right side until a triangle was selected on the left and all triangles touch only one Edge segment (after swapping).

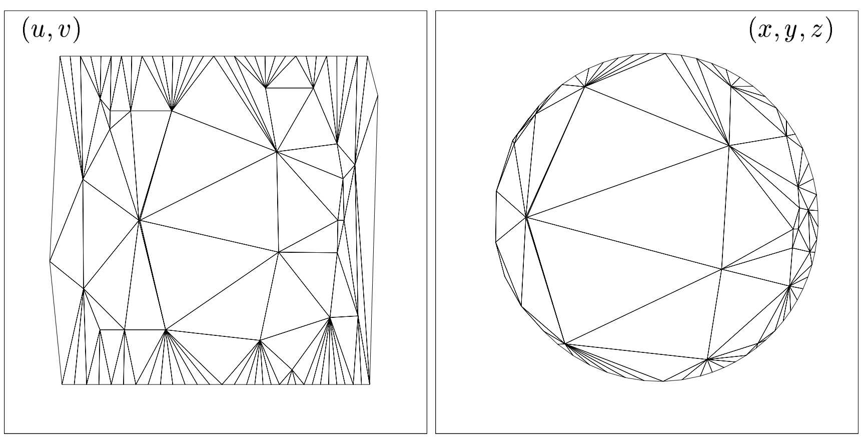

Figure 8: Phase 3 – Triangle subdivision where the midpoints do not map. # points = 90, # triangles = 114, # CAD queries = 176.

The resultant triangulation has the property that any triangle side whose vertices touch more that a single Edge are those that are on the CAD Edge.

3. Subdivide until consistent. Triangle subdivision is performed anyplace that the triangle centroid in (u, v) does not map into the triangle in physical coordinates. See Figure 8. At this crucial step, the triangulation is "consistent" in both (u, v) and (x, y, z) correct though at low fidelity. At the end of this phase, the physical space triangulation is a good approximation of the underlying CAD geometry. This triangulation provides a basis for more refinement and further improvements may be performed in any order.

2.3 Mesh Improvement

The improvement phases used are listed below. They are presented in isolation, but the actual algorithm has the ability to use all (sequentially). The phases are described in the order that have been implemented but, in fact, could have been used in any order (although it is best to use Phase 7 last).

4. Maximum side length. Starting with the "consistent" triangulation, from Phase 3, sides are split where the length is greater than the prescribed value. Figure 9 shows the result after the phase is complete. The regularity in the triangulation is noteworthy as is the relatively small number of CAD queries.

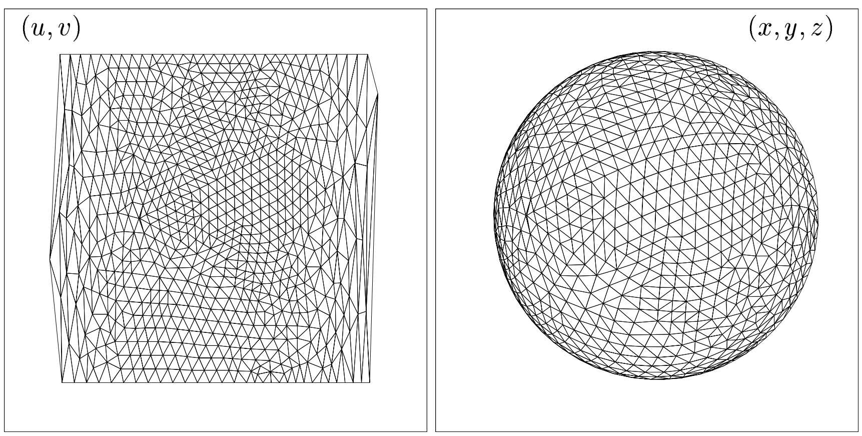

Figure 9: Phase 4 – Triangle sides split that are greater 15% of the radius. # points = 736, # triangles = 1406, # CAD queries = 882.

5. Triangle dihedral angle. Triangle subdivision is performed where the normal vectors of neighboring facets differ by more than a specified angle value. When a pair is found, the triangle with the larger (x, y, z) area is subdivided. Figure 10 displays the result starting directly with the consistent triangulation seen in Figure 8.

6. Chord-height tolerance. Triangle subdivision is performed when the deviation from the center of the triangle in (u, v) is greater than the prescribed

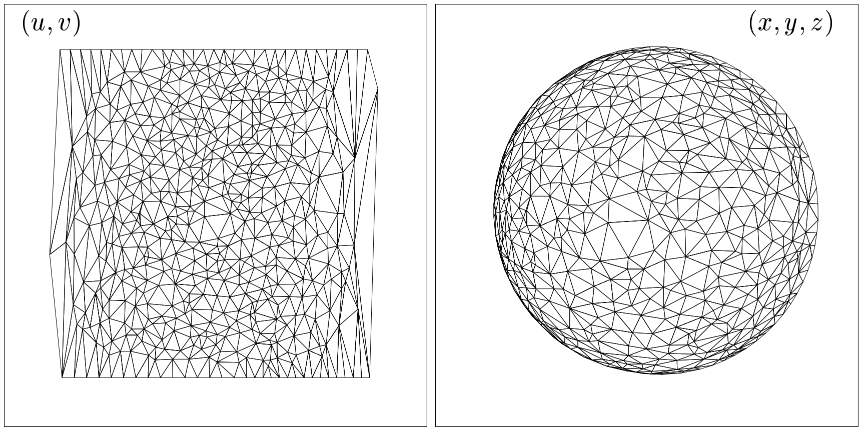

Figure 10: Phase 5 – Triangle subdivision when facet dihedral angles are $> 8^o$. # points = 531, # triangles = 996, # CAD queries = 3141.

distance. Note this is a different criterion than that originally desired. The chordal deviation at the triangle center in (x, y, z) would require a snap to the surface (which we are avoiding). The comparison is done at the triangle center in parameter space and can be performed by a simple evaluation. The larger number of CAD queries is required; at least one for each triangle constructed during the entire phase. In Figure 11 some clustering of vertices exists near the poles due to the distortion in mapping around these areas.

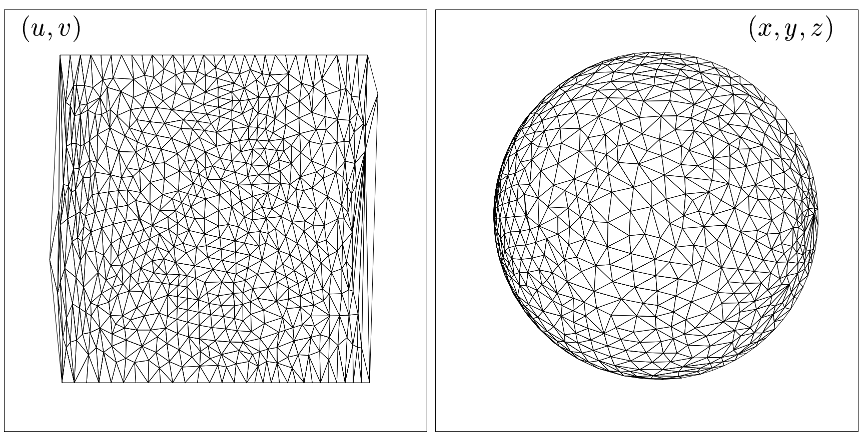

Figure 11: Phase 6 – Subdivide with centroid deviation $> 1/2\%$ R. # points = 621, # triangles = 1176, # CAD queries = 3591.

7. Surface recovery minimizing the (x, y, z) angles. This is seen in Figure 12 for the result of Phase 4 only. The reconstruction is also constrained so that the overall quality is not diminished.

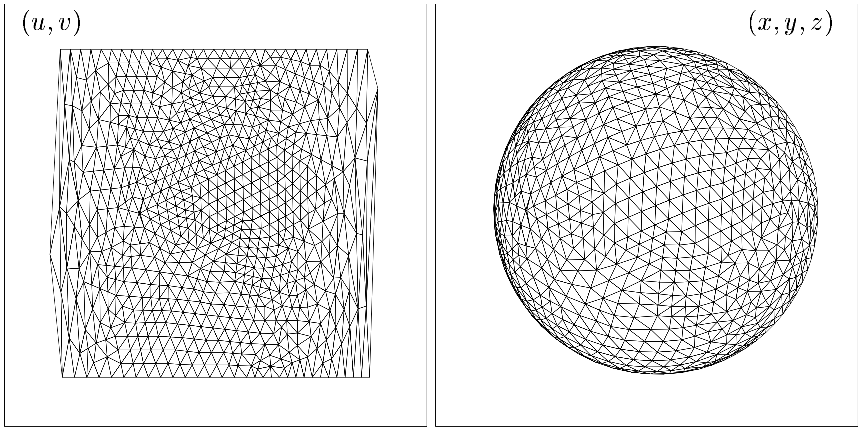

Figure 12: Phase 7 – Surface recovery minimizing (x, y, z) angles starting from Figure 9. # points = 736, # triangles = 1406, # CAD queries = 882.

3 Edge Discretization

During the work on the surface tessellation algorithm, it became apparent that the only way to insure a high fidelity triangulation is to have "good" Edge discretizations. It is easy to see that if an Edge segment is longer than the maximum triangle side length then the tessellation cannot obtain the desired criterion because these sides are not split. This is true for any criteria.

An Edge procedure was constructed consistent with the Face tessellation algorithm. The phases for this procedure follow:

- Start from the bounding Nodes.
- Bisect the line segment in t if the Edge is periodic or there are only 2 Edges within an attached Face Loop.
- Bisect segments in t where the segment length is greater than the maximum permitted edge length.
- Bisect segments where the angle between point tangents is larger than the maximum allowed dihedral angle.
- Bisect segments when the deviation from the center in t is greater than the chord-height tolerance.
- Bisect segments where there is a high dihedral angle between the attached Faces. Consider a wing where there are two defining surfaces that are split down the leading edge. The leading edge is a straight line so the result of the above options is that no points are inserted (if the maximum triangle side specification is not used). With one large segment it is impossible to refine the tessellation close enough to the leading edge to properly follow the curvature. The technique used in this phase is as follows (see Figure 13):

    1. Midpoint $t \rightarrow (x, y, z)_m = g(t)$
    2. $(u, v)_m = f^{-1}(x, y, z)_m$

3. Get surface normal at $(u,v)_m$
4. Cross normal with the segment vector, 1/4 of the segment distance sets $(x,y,z)_p$
5. $(u,v)_p = f^{-1}(x,y,z)_p$
6. Get surface normal at $(u,v)_p$
7. Insert the midpoint if the angle between the normals at m and p is greater than the maximum prescribed value

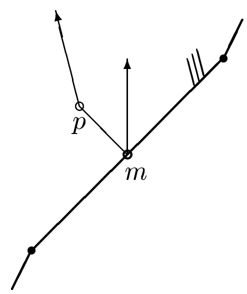

Figure 13: Surface curvature based Edge discretization enhancement

- Bisect the larger segments when the length changes by more than a factor of 4.

It should be noted that t's are used to set the points in physical space. In order to perform the Face tessellation algorithm (u,v) coordinates are required for each Edge vertex for the associated Faces. Therefore snaps, $f^{-1}(x,y,z)$, are required to get the coordinates on the surfaces in parameter space. By evaluating the resultant (u,v)s, from the snap, a distance between the Edge vertex and the position on the surface can be obtained. As discussed earlier, snaps performed by many CAD systems can return untrustworthy values, the value returned by a snap is treated with suspicion and vertices are rejected if the distance fails any of the following tests:

- Segment spacing $< 10^{-4}$ reference distance
- Distance between Edge (x,y,z) and $f(u,v) >$ segment length
- Distance between Edge (x,y,z) and $f(u,v) >>$ other segments

When snapping points onto the surfaces, special care must be taken with:

- Periodic parameterization. Is the result in the correct quadrant?
- Mapping degeneracies. At times, the snap returns *strange* positions. This is especially true near singularities in the mapping. Also, loops may not appear to be closed in parameter space.
- Edge crossovers. When two consecutive Edges mate at a shallow tangent, and both display high curvature, it is possible that the segments may cross. This is due to the discrete nature of the curves and the point selection technique.

4 Discussion

Figure 14: Overview of an X29-like aircraft showing the CAD Face on the fuselage used in the following examples.

To provide a better understanding of the behavior of each of the three mesh improvement techniques, consider the aircraft geometry shown in Figure 14. This geometry depicts an X29-like aircraft, with a CAD face near the rear of the fuselage highlighted. Triangulation of this CAD Face is instructive since, unlike the earlier sphere example, the surface is both anisotropic and non-uniform. In examining these improvements, we track the number of triangles, vertices, and CAD queries to show the relative cost of the various improvement techniques.

Figure 15 depicts the consistent triangulation after Phase 3 of the process described in section 2.2. At this point, the surface distortion has been reduced such that each triangle centroid in (u, v) maps into the same triangle in (x, y, z). This triangulation forms the basis for the "improvements" which follow. The next three figures show the effects of refining this triangulation based on maximum allowable edge-length, dihedral angle, and chord-height.

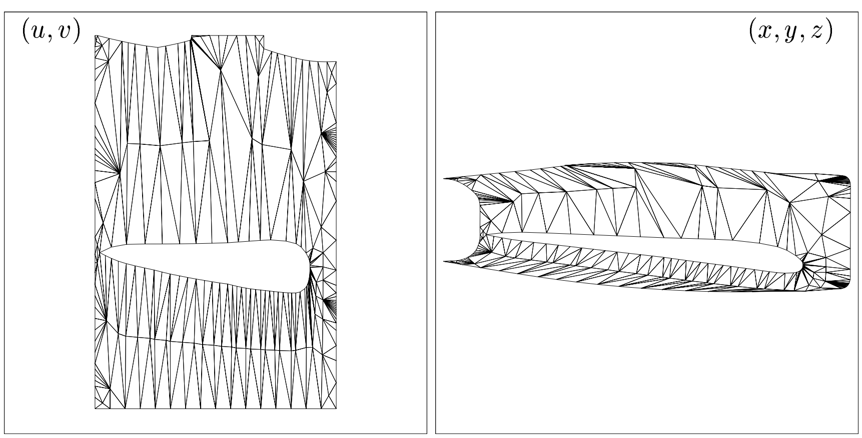

Figure 15: Phase 3 – Consistent triangulation; each triangle centroid in (u, v) maps into the same triangle in (x, y, z). # points = 257, # triangles = 321, # CAD queries = 392.

Side-length refinement: Figure 16 shows the result after the triangle sides are split such that no segment exceeds the maximum permitted length. It should be noted that at this point the surface is well developed and the procedure only required about as many CAD queries as there are triangles in the Face tessellation.

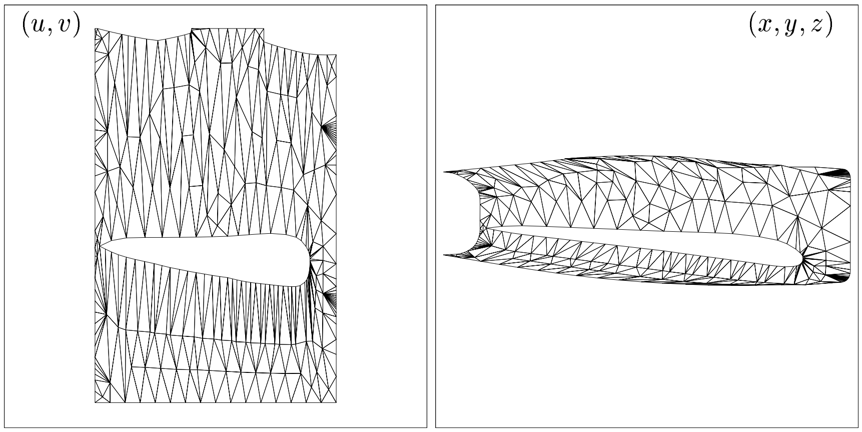

Figure 16: Phase 4 – Triangle sides split when length is > a specified value. # points = 312, # triangles = 431, # CAD queries = 447.

Dihedral Angle refinement: Figure 17 displays the result of improving the consistent mesh (Figure 15) when the adjacent facet angles are larger than 10^o. Since this geometry has anisotropic and non-uniform curvature, the total number of CAD queries and the total number of triangles generated are quite large. Nevertheless, the tessellation meets the specified dihedral angle criteria.

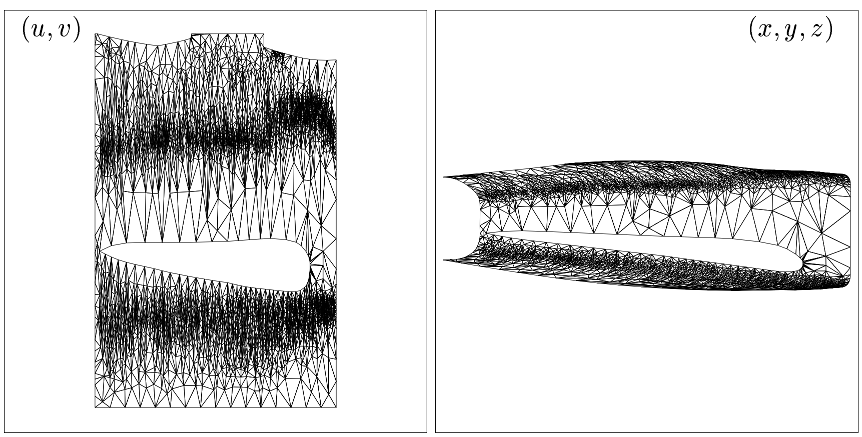

Figure 17: Dihedral angle refinement on the consistent triangulation of Figure 15 using a 10^o maximum allowable deviation between adjacent triangles. # points = 3168, # triangles = 6143, # CAD queries = 20642.

Chord-height refinement: Finally, Figure 18 depicts triangulation after enforcing a chord-height tolerance on the consistent triangulation of Figure 15. Triangles were subdivided anytime the centroid of the triangle in (u, v) was greater than the prescribed distance from the underlying CAD geometry. The cost of this refinement technique is approximately three CAD queries for each triangle in the final mesh. Substantially fewer triangles were generated using this improvement mechanism than that of the last figure.

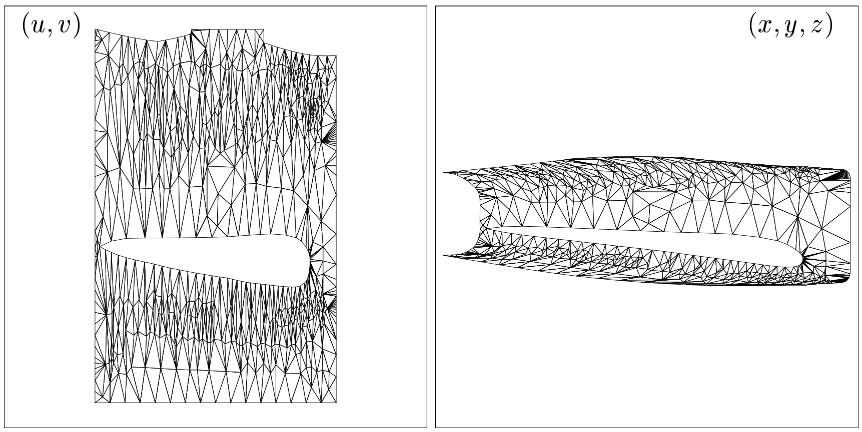

Figure 18: Chord-height based refinement of the consistent triangulation seen in Figure 15. # points = 600, # triangles = 1007, # CAD queries = 2915.

In examining the previous three figures, its clear that the dihedral angle based refinement is sensitive to anisotropic curvature in the surface. Since the simple subdivision rules do not attempt to align inserted vertices with the direction of principle curvature, surface bends, cylindrical trailing edges, fillets and the like, all require a large number of triangles to satisfy the angle metric. When combined with the isotropic nature of the MinMax triangulation in (u, v), this misalignment can make dihedral-angle based refinement expensive.

Preliminary timing analysis indicates that as the number of triangles increases on a Face, the proportion of time spent swapping for surface recovery grows rapidly. While triangulation speed can be improved through the use of recursive swaps and priority queues, many triangles will still be required at anisotropic surface features without a more sophisticated site insertion strategy.

Figure 19 depicts one half of the X29-like aircraft where all 34 CAD Faces were tessellated with the same improvement parameters. For this example all phases were used. One can clearly see the transition from side-length refinement to curvature improvements. A dihedral angle refinement parameter of 12^o was used. This tended to displace any chord-height improvements for the value of the parameter used, as a result, relatively few points were inserted in Phase 6.

Note that many of the junctures between CAD Faces are not obvious. This is due to the consistent manner that both the Edge and Face discretizations were performed.

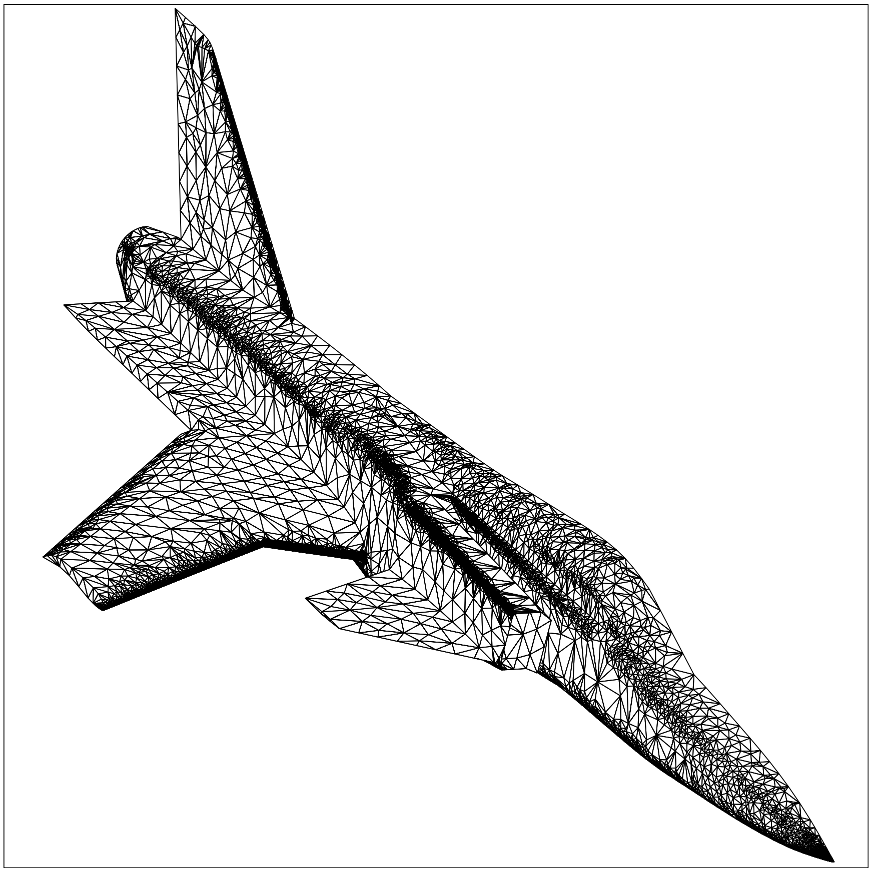

Figure 19: Complete tessellation of the X29-like aircraft.

5 Conclusions

This paper presents a technique for generating high quality water-tight triangulations directly from CAD in an independent, vendor neutral setting. The robust scheme works for *real* solid geometries, even those that are considered *dirty* and inconsistent – candidates that would require manual "CAD repair" in almost any other setting.

The approach first performs a boundary triangulation, then by using simple triangle insertion operations, points are added on the surface until the mapping in parameter space is consistent with physical space. A number of user adjustable mesh improvement strategies were also discussed. These included those for meeting triangle side length, dihedral angle and chord-height requirements.

Although the algorithm described here works well, some improvement is still possible. The aircraft fueslage example demonstrated that anisotropic curvature may introduce a large number of triangles under certain improvement schemes. Future work may involve effectively dealing with these situations by making an effort to align site insertions and edges normal to the primary curvature. In addition, now that there is something that produces the desired result, general performance enhancements can be researched and applied.

Acknowledgments

This work was partially sponsored by NASA Glenn Research Center (NAG3-2523) with Gregory Follen as Technical Monitor and NASA Ames Research Center (NAG2-1458) with Goetz Klopfer as the Technical Monitor.

References

[1] R. Haimes, and G. Follen. **C**omputational **A**nalysis **PR**ogramming **I**nterface. Proceedings of the 6^{th} International Conference on Numerical Grid Generation in Computational Field Simulations. University of Greenwich, June 1998.

[2] E.A. Gerteisen and T.J. Baker. Meshing of Trimmed Surfaces by Adaptive Refinement. Proceedings of ECCOMAS 2000, September 2000.

[3] G. Farin. *Curves and Surfaces for Computer Aided Geometry Design. A Practical Guide*, Academic Press, 1990.

[4] L.P. Chew. Guaranteed-Quality Mesh Generation for Curved Surfaces. Proceedings of the Ninth Annual Symposium on Computational Geometry, pp. 274-280. ACM, 1993.

[5] M. Aftosmis, M. Delanaye and R. Haimes. Automatic Generation of CFD-Ready Surface Triangulations from CAD Geometry. AIAA Paper 99-0776, January 1999.

[6] J. O'Rourke. *Computational Geometry in C*, Cambridge Univ. Press, 1993.

[7] O. Hassan, N.P. Weatherill, J. Peraire, K. Morgan, and E.J. Probert. Generation and Adaptation of Unstructured Meshes. COSMASE Course on Grid Generation, CERFACS, Feburary, 1995.